

APPLICATION OF 1553B TO MRASM-A SYSTEMS LOOK

John E. Leib
General Dynamics Convair Division
P.O. Box 80847
San Diego, California 92138
(714) 277-8900, x2101

ABSTRACT

MRASM, is the first missile to incorporate a MIL-STD-1553B data bus as the primary means of data transfer among the elements of the missile. The Standard, built around applications which could dedicate major computing power to manage the affairs of the data bus, posed a challenge to MRASM because this bus management function needed to be performed on the input/output card which fit in an existing computer design, while not utilizing its host computer on a continuing basis.

This paper reviews the process by which the data bus operation was defined, describes the protocol adopted for timely transfer of data, and argues the case for the system design decisions.

INTRODUCTION

The Medium Range Air to Surface Missile (MRASM) is an adaptation of Tomahawk for air carry and launch. For use by both the Air Force and the Navy, the MRASM must accommodate various configurations of payloads and avionics. A MIL-STD-1553B data bus was selected for data transfer among the many computers in the system, to aid in easy integration through use of a standard, widely used and understood technique.

Applying MIL-STD-1553B to this missile was a "first", so there was little history to guide the system designer as he groped for the proper definition to best serve the MRASM program. Severe space, weight and power restrictions made the luxury of committing large amounts of computing capability to support the operation of the data bus an unlikely solution; and some of the terminals were required to interface with existing computer designs which had been selected for use on MRASM.

Latency of some data could be critical, as MRASM is basically an unstable vehicle during some portions of flight, and requires a tight autopilot loop. And, as always early in the definition of a data bus, bus loading, or duty cycle was of concern.

The processes and decisions which resulted in defining the requirements for the MRASM internal MIL-STD-1553B data bus follows.

1.0 MRASM Hardware Configuration

The basic MRASM has five boxes (Figure 1) which communicate with each other over the data bus. These are:

- MCM - Mission Control Module
- GNC - Guidance and Navigation Computer
- ISA - Inertial Sensor Assembly
- DPU - DSMAC Processor Unit
- SMC - Stores Management Controller

In addition, on test flights there is a

- TIC - Test Instrumentation Controller

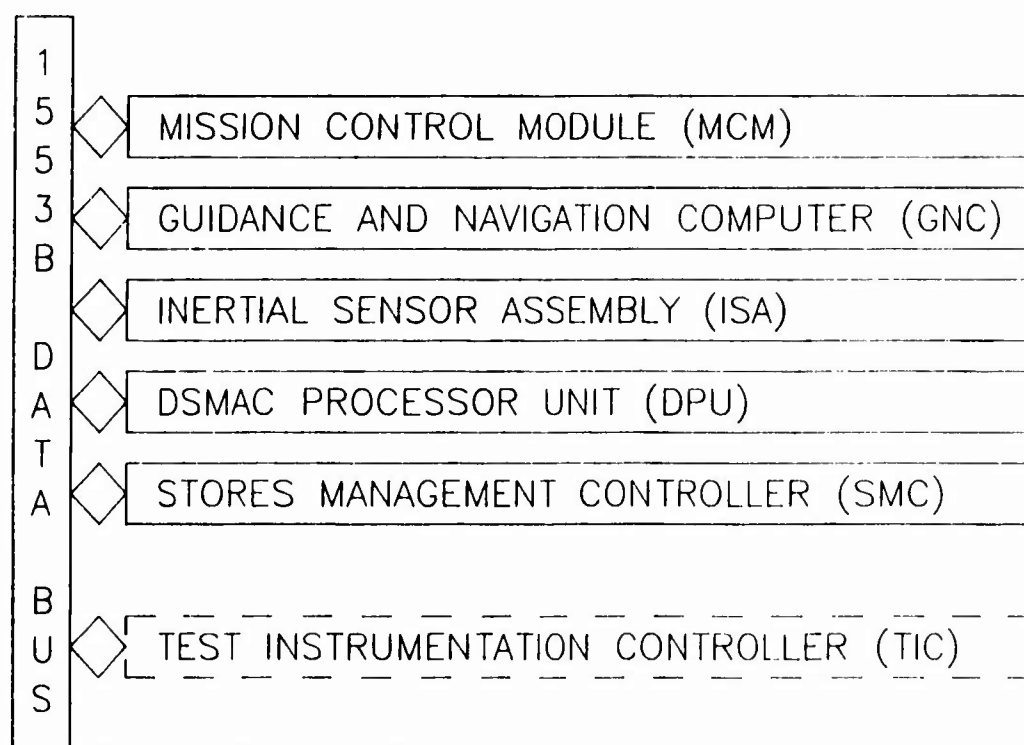


Figure 1. MRASM Data Bus Configuration

The MCM performs the sequencing and autopilot functions, and is a Digital Integrating Subsystem (DIS) computer. It receives guidance and throttle commands from the GNC and inertial data from the ISA, and provides air data to the GNC, DSMAC scene data to the DPU, and payload dispensing data to the SMC.

The GNC performs guidance and navigation computations, and is also a DIS computer. It receives inertial data from the ISA, air data from the MCM, and update data from the DPU; and provides guidance and throttle commands to the MCM.

The ISA contains gyros and accelerometers, and measures its inertial environment and makes some calculations to perform the "gimbal" functions. It sends autopilot data to the MCM, navigation data to the GNC, and attitude data to the DPU.

The DPU processes camera data and compares it with stored scenes to determine where it is. It receives the stored scenes from the MCM and sends its position match data to the GNC.

The SMC controls the dispensing of airfield attack payload submunitions. (For other payloads, it may not be aboard.) It receives payload dispensing commands from the MCM. It does not transmit any messages.

All of this interchange of data is carried out on a MIL-STD-1553B data bus. (Some other data moves on other mediums, but none of it is "computer to computer".) So the first question to be answered was, "By what specific protocol was this data to be made to move?". Constraints on the answer included the fact that software for the various computers was being developed by several independent entities and, while there is some rough synchronization among the computing activities, the precise time a computer would have a message ready for transmission was not known to any other computer. Also, the press of other activities made it unlikely that any of these computers could take on the additional task of truly managing the affairs of the bus in a real-time sense: the bus interfacing hardware must also perform this function.

On test flights, the TIC, also a DIS computer, collects and formats data for telemetry. Most of this is extracted from traffic on the data bus, the TIC acting as a monitor terminal. This aspect was not a factor in considering the bus protocol, but did contribute its share of requirements for the hardware implementation of the bus.

2.0 Protocol Options Evaluated

Within MIL-STD-1553B, three protocol options were identified as candidates for implementation in MRASM. These were called (1) Command/Response, (2) Passing Protocol, and (3) Poll for Transmission, and were considered to be the only technically viable possibilities within the Military Standard.

2.1 Command/Response

This protocol might be viewed as basic MIL-STD-1553B. The Bus Controller must know the schedule by which each message is prepared to be sent and appropriately command the transmission and reception of that message. (The word "message", as used in this paper, means a group of data words carrying functional information among the computers, and not a group of Command, Data and Status words, as defined in the Standard.)

2.2 Passing Protocol

Using this protocol, each terminal which may have a message to transmit is placed in the sequence of Bus Controllers, and control is passed in this sequence using the Dynamic Bus Control mode code command. Upon being designated "Bus Controller", a terminal transmits a message if one is ready, then passes control to the next terminal in the sequence. If no message is ready, control is passed immediately.

2.3 Poll for Transmission

This protocol requires a sequence of terminals to be known to the Bus Controller. (Any terminal may appear more than once in the sequence if higher frequency access is required.) The Bus Controller "polls" the terminals in this sequence for messages ready for transmission. If a terminal notifies the Bus Controller of a ready message, the Bus Controller provides the necessary command words to cause that message to be both transmitted and received. Of course, the Bus Controller must also be in this sequence and its messages are output during its turn.

2.4 Evaluation Arguments and Selection

MRASM being a tactical weapon which will be built by the thousands, total vehicle cost was a major consideration. Ease of integration among several contractors' equipments, and flexibility to add and delete equipment with minimum impact were also significant selection factors. While technical adequacy was mandatory, this did not appear as a serious threat to any of these candidates and was not a discriminator in the selection.

2.4.1 Evaluation of Command/Response

The advantage offered by the Command/Response protocol is that it is the most applied, and therefore the most familiar use of MIL-STD-1553B. The Bus Controller must know what is happening in all the computers on the bus in enough detail to know when a new message has been constructed and placed in an output buffer, how to address the output buffer, which terminal or terminals should receive this message, and how to direct the message to the proper input buffer or buffers. A significant, dedicated computational capability is required to support the Bus Controller in managing the data transfer on the bus, and tight synchronization among computers is required to avoid large latencies.

Because of this intimate, all-knowing involvement with every message on the bus, the Bus Controller must be altered in some way whenever there is an addition, deletion or change of any message.

MRASM is a system built from components from many contractors, and the integration problem for Command/Response protocol would be formidable, indeed. Not only would each message need to be agreed to by the sender and receiver, but much accurate information would also need to be incorporated into the operation of the Bus Controller.

2.4.2 Evaluation of Passing Protocol

The integration process becomes easier with Passing Protocol than with Command/Response. This is achieved because the messages and the protocol are decoupled to a great extent. No synchronization requirements are imposed simply to make the protocol work. Each computer determines when it has a message for transmission, then sends it when it is ready to send it. No other terminal needs to know when this will be. Additions, deletions and changes of messages are negotiated between the sender and receiver with no impact on the bus management function.

The bus management function at each terminal does need to know the next terminal in the sequence so that control can be passed properly. This imposes some attention to detail when terminals are either added to, or removed from the sequence, but this represents a major change in the vehicle configuration compared to changing or restructuring the messages on the bus, anyway.

A disadvantage of this protocol is that every terminal must be capable of becoming the Bus Controller, with its attendant added complexity.

2.4.3 Evaluation of Poll for Transmission

The Poll for Transmission protocol combines all of the benefits of Passing Protocol with the added benefit of allowing all but one terminal to be a Remote Terminal. The only bus-management-peculiar data required by the Bus Controller is the polling sequence.

The disadvantage of this protocol is that there is added protocol which increases bus loading. The significance of this, or lack of significance, is determined by the application. For MRASM, this was not considered to be very important.

2.4.4 Protocol Selection

For MRASM, the Poll for Transmission protocol was selected because it minimizes total program cost, integration complexity and attendant problems, and the impact of message additions, deletions and changes.

The specific protocol calls for the Bus Controller to "poll" each Remote Terminal in the selected sequence by addressing a Transmit Vector Word mode code command to that Remote Terminal. The Remote Terminal responds with a Status Word and a Vector Word, in accordance with MIL-STD-1553B. If that terminal has a message to be transmitted, the Service Request bit in the Status Word is set, with the data in the Vector Word supplying the information needed by the Bus Controller to cause the transmission and reception of the Remote Terminal's message. If the Service Request bit is not set, the Vector Word is ignored.

Figure 2 shows this process for an RT-to-RT message transfer. For RT-to-BC and RT-to-Broadcast transfer, the protocol following the Vector Word is adjusted in accordance with the Standard.

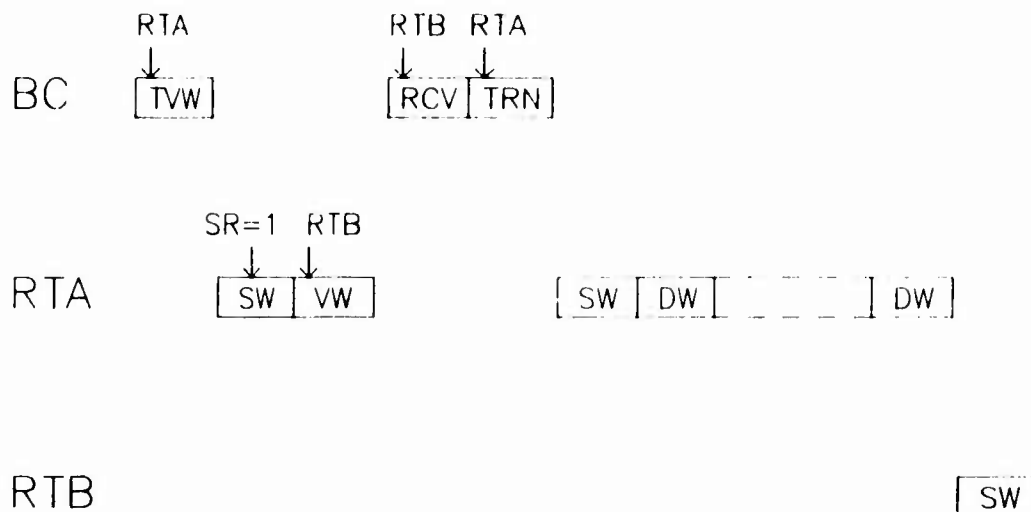


Figure 2. Poll for Transmission Sequence

When the Bus Controller has a message to send, it waits for its turn in the polling sequence, then issues a receive command followed by the message, also in accordance with the Standard.

3.0 Bus Implementation Detailed Requirements

Since two of the MRASM computers are from the DIS family (and a third computer on test flights), there was a requirement to supply one computer with a MIL-STD-1553B input/output channel which would act as a Bus Controller, and one computer with a Remote Terminal, each to fit into the standard DIS I/O slot. The third, flight test computer needed a Monitor Terminal. The detailed requirements included these facts, and the challenge for the I/O card designer was further elevated by the requirement that a single hardware/firmware design would act as all three, the specific type of terminal being selected by software in the host computer.

Several options are provided in the Standard, and from these options were selected the requirements for the specific design, as viewed from the bus. The other sets of requirements were dictated by the need for compatibility with the existing I/O card slots in the DIS computers, and by MRASM system considerations.

3.1 Bus Oriented Requirements

The requirements of MIL-STD-1553B were imposed on the card design. Options and alternative selections permitted by the Standard are described here.

3.1.1 Subaddress/Mode Field of Command Words

The Subaddress/Mode field is used in the subaddress application with the first bit (MSB) of the field set to one and the second bit set to zero. The third bit is used to indicate a "priority" message when it is set to one. In DIS computers, priority messages are handled entirely by the operating system. The fourth and fifth bits are set to zero for directed messages, and used to indicate the "broadcast group" to which a broadcast message belongs. Each DIS computer will receive one or more broadcast groups if it receives any broadcast messages.

For mode codes, only "11111" is used. Thus the first bit in this field will always be set to one, and may be used to distinguish command words from status words, which have a zero in this location (the "Instrumentation Bit").

3.1.2 Mode Codes

The only mode code required to be implemented is the Transmit Vector Word mode code. In response to this mode code command, a Remote Terminal will indicate the availability of a message by setting the Service Request bit in its Status Word to one, and providing a vector word which indicates, in Command Word format, which terminal or terminals the message is for, whether it is a priority message, and how many data words the message contains. Except for messages directed to the Bus Controller, the Bus Controller places the vector word on the bus with command word sync (it will be a "receive" command), followed by a "transmit" command directed to the "polled" terminal, with the last ten bits identical to the corresponding bits in the Vector Word.

All other mode codes are not used in the MRASM protocol and their implementation is optional.

3.1.3 Cable Stub Requirement

The requirements for direct coupled stubs, as described in 4.5.1.5.2 of the Standard, apply to the card design.

3.2 DIS Host Computer Oriented Requirements

In addition to interfacing with the data bus, the card must interface with the host computer. These are the major requirements imposed on the card by this interface.

3.2.1 Dimensions and Form Factor

To fit into a DIS computer, the card must be designed on one side of a DIS standard printed circuit board with a compatible, 70-pin connector. The major implication of this is that there are only about twenty square inches on which to fit the components.

3.2.2 Power

For the survival of the card and the DIS computer, the average power is limited to 10.5 watts, with peaks of no more than 13 watts.

3.3 MRASM System Oriented Requirements

MRASM system requirements were leveled on the card in great numbers. These requirements were those which would be necessary to make the system work. The major ones which drove the design are described here.

3.3.1 Card Reconfiguration

As the card is to perform as a Bus Controller, a Remote Terminal, or a Monitor Terminal, selectable at will by the host computer, a method for providing this information was developed and defined. Upon a signal from the host computer, the card extracts this information from the host computer memory, using its direct memory access channel. This information provides everything necessary for the card to know how to properly perform. Called the "reconfiguration message", it is composed of twenty 16-bit words.

3.3.2 Input Procedures

A problem sometimes encountered in a data bus implementation is overwriting data before it has been completely processed by the receiving computer. A similar problem occurs when two or more messages arrive before the first interrupt announcing their arrival can be honored by the host computer, and all but the last of the messages are lost.

For MRASM, messages are stored in sequentially-identified buffers in memory. This allows the host computer to handle all incoming messages at its own pace.

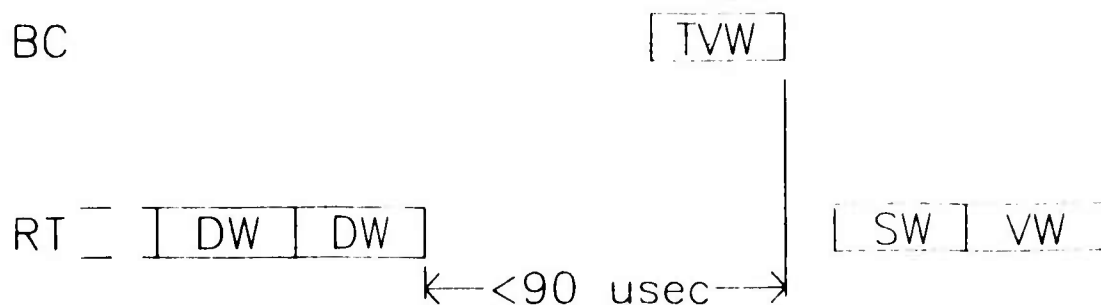
3.3.3 Message Retries

The requirement to assure the correct transmission and reception of every message falls to the Bus Controller. If the Bus Controller determines this has not occurred, it will initiate a "retry", and will continue to do so until it determines the message has been successfully transferred or it has initiated the number of retries called for in the reconfiguration message.

Remote Terminals must be prepared to support these retries, both as senders and receivers. A retry is commanded by issuing a repeat of the previous Transmit Vector Word mode code command within 90 microseconds after the final data word of a message has been placed on the bus (Figure 3).

3.3.4 Reliability

The MRASM requirement for card reliability at 75°C is 45000 hours MTBF.



A "Transmit Vector Word" mode code command received less than 90 u-sec after last data word is transmitted initiates a "Retry" of same message

Figure 3. Hardware Retries

4.0 Bus Performance

Concern for bus loading and data latency has resulted in close monitoring of estimates of these quantities as the bus traffic is being defined. An analysis program was developed to calculate loading and latency, with bus traffic the input. As these estimates have become more firm, the concern appears to be more weakly founded.

Present bus loading estimates for operational flights (without the TIC) are under 16% during the busiest phase of flight, and less than 21% with the TIC on test flights. Bus latency averages about one millisecond, with worst-case latency conservatively estimated at three milliseconds.

REFERENCES

- 1) MIL-STD-1553B, dated 12 February 1980.
- 2) General Dynamics Convair MRASM document No. 109-DRB-4005, Revision C, dated 19 August 1982.

JOHN E. LEIB

Mr. Leib has 29 years experience in technical and management assignments at General Dynamics Convair Division. These assignments have been in design and analysis of large systems, including the Atlas and Centaur space boosters and the Tomahawk cruise missile. Currently, he is an Engineering Staff Specialist, conducting and directing system definitions and analyses for the Medium Range Air to Surface Missile, and was one of the architects of the application of MIL-STD-1553B to the internal communications data bus of the MRASM.

Mr. Leib received the BSEE from the University of Michigan in 1951, and the MSEE from the California Institute of Technology in 1953.

